



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Group Art Unit: 2124

Examiner: Unassigned

In Re PATENT APPLICATION Of:

Applicants : Kimito HORIE)

Serial No. : 10/648,373)

Filed : August 27, 2003)

For : INTEGRATED CIRCUIT)

Attorney Ref. : OKI 372)

**SUBMISSION OF
PRIORITY DOCUMENT**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith is a certified copy of applicant's first-filed Japanese Application No. 2002-246629, filed August 27, 2002, the rights of priority claimed on August 27, 2003 with the filing of the above-referenced application.

It is respectfully requested that receipt of this priority document be acknowledged.

Respectfully submitted,

Robert H. Berdo, Jr. (Reg. No. 38,075)
RABIN & BERDO, P.C.
(Customer No. 23995)
Telephone: (202) 371-8976
Telefax: (202) 408-0924

December 17, 2004
Date

RHB:pjl

FEE ENCLOSED:\$
Please charge any further
fee to our Deposit Account
No. 18-0002

日 本 国 特 許 庁
JAPAN PATENT OFFICE

CERTIFIED COPY OF
PRIORITY DOCUMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 8月27日

出 願 番 号

Application Number:

特願2002-246629

ST.10/C]:

[JP2002-246629]

出 願 人

Applicant(s):

沖電気工業株式会社

2003年 2月14日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2003-3006947

BEST AVAILABLE COPY

【書類名】 特許願

【整理番号】 KT000450

【提出日】 平成14年 8月27日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 G06F 7/50

【発明者】

 【住所又は居所】 東京都港区虎ノ門1丁目7番12号 沖電気工業株式会
社内

 【氏名】 堀江 公人

【特許出願人】

 【識別番号】 000000295

 【氏名又は名称】 沖電気工業株式会社

【代理人】

 【識別番号】 100095957

 【弁理士】

 【氏名又は名称】 亀谷 美明

 【電話番号】 03-5919-3808

【選任した代理人】

 【識別番号】 100096389

 【弁理士】

 【氏名又は名称】 金本 哲男

 【電話番号】 03-3226-6631

【選任した代理人】

 【識別番号】 100101557

 【弁理士】

 【氏名又は名称】 萩原 康司

 【電話番号】 03-3226-6631

【手数料の表示】

 【予納台帳番号】 040224

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9707549

【包括委任状番号】 9707550

【包括委任状番号】 9707551

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 加算器、乗算器、及び集積回路

【特許請求の範囲】

【請求項 1】 第 1 と第 2 の 2 つのデータ入力と、第 1 と第 2 の 2 つのキャリ入力と、キャリ選択入力とを入力とし、

前記第 1 と第 2 の 2 つのデータ入力の X O R 出力を生成する第 1 の X O R 素子と、

前記 X O R 出力を選択信号として前記第 1 のキャリ入力と前記第 1 のデータ入力とのいずれかを選択する第 1 のマルチプレクサと、

前記第 2 のキャリ入力と前記第 2 のデータ入力とのいずれかを選択する第 2 のマルチプレクサと、

前記キャリ選択入力を選択信号として前記第 1 と第 2 の 2 つのキャリ入力のいずれかを選択する第 3 のマルチプレクサと、

前記第 3 のマルチプレクサの出力と前記 X O R 出力との X O R 出力を生成する第 2 の X O R 素子とを有し、

前記第 1 のマルチプレクサの出力を第 1 のキャリ出力とし、

前記第 2 のマルチプレクサの出力を第 2 のキャリ出力とし、

前記第 3 のマルチプレクサの出力を加算値としたことを特徴とする加算器。

【請求項 2】 請求項 1 に記載の加算器が複数段継続接続され、

前段の第 1 のキャリ出力を後段の第 1 のキャリ入力とし、

前段の第 2 のキャリ出力を後段の第 2 のキャリ入力とし、

前記複数の加算器のキャリ選択入力を全段で共通とし、

初段の加算器の真のキャリ入力を前記キャリ選択入力とし、第 1 のキャリ入力を第 1 の仮想キャリとし、第 2 のキャリ入力を第 2 の仮想キャリとし、

前記キャリ選択入力により最終段の加算器の第 1 と第 2 の 2 つのキャリ出力のいずれかを選択する第 4 のマルチプレクサを有し、

前記第 4 のマルチプレクサの出力をキャリ出力としたことを特徴とする加算器

【請求項 3】 前記初段の加算器の前記第 3 のマルチプレクサを省き、前記

初段の加算器の前記第 3 のマルチプレクサの出力の替わりに前記キャリ選択入力を用いることを特徴とする請求項 2 に記載の加算器。

【請求項 4】 請求項 2 または請求項 3 のうちのいずれか 1 項に記載の加算器を複数個継続接続し、前段のキャリ出力を後段のキャリ入力とする構成の加算器であって、

最終段の加算器を除き、各段の加算器の処理可能なビット数は、その 1 つ前段の加算器の処理可能なビット数と等しいか、または大きくしたことを特徴とする加算器。

【請求項 5】 初段の加算器を除き、各段の加算器の処理可能なビット数と、その 1 つ前段の加算器の処理可能なビット数との差が一定であることを特徴とする請求項 4 に記載の加算器。

【請求項 6】 ツリー構造で構成されるブース乗算器であって、最終段の加算器に、請求項 4 または請求項 5 に記載の加算器を用いたことを特徴とするブース乗算器。

【請求項 7】 請求項 1, 2, 3, 4, または 5 のうちのいずれか 1 項に記載の加算器を含む集積回路。

【請求項 8】 請求項 6 に記載の乗算器を含む集積回路。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、加算器、乗算器、及びこれらのいずれかを用いた集積回路に関し、特に、従来のキャリセレクト方式を改良して回路量を減らし、かつ、高速化した加算器、乗算器、及びこれを用いた集積回路に関する。

【0 0 0 2】

【従来の技術】

近年の DSP (Digital Signal Processing) の発達はめざましく、その中核技術として 32 ビット乗算器等が採用されている。その構成には全加算器を N^2 個用いた単純な乗算器は採用されず、多くは 2 次ブース乗算器 (Booth Multiplier) が採用される。加算器を経由す

るキャリ (Carry) の遅れが大きいことから、部分積 (ブース積; Booth Muxes) の個数を減らして、回路量の削減や乗算速度の向上を追求するためである。部分積の寄せ集めについては、CPA (Carry Propagation Adder) を用いる方法もあるが、キャリの遅れを考慮して、ワレスツリー (Wallace Tree) を採用することが多い。

【0003】

ワレスツリーは、CSA (Carry Save Adder) のツリー構造で構成される。CSAはキャリ別加算を行うのでキャリの伝搬が起こらず、キャリの遅延も少なく済む。しかし、寄せ集めたブース積は最終段で65ビット加算器により通常の加算が行われるので、その部分のキャリの遅れが乗算器の高速化のネックになっていた。また、64ビット乗算器においても129ビットの高速加算器が要求されるため、このことが64ビット乗算器を実現する際の障害になっていた。

【0004】

この事情は暗号の分野では特に深刻である。暗号の秘匿性を向上させるためには多ビットの加減乗除を行う必要があり、場合により剰余演算も必要である。今日では、RSA方式の暗号では1024ビットの数を、楕円曲線暗号方式では224ビットの数を扱う必要がある。暗号を生成し、若しくは復号し、更に署名やその検証を行うためには、このような多ビットの計算を高速で行わなければならない。

【0005】

加算器を高速化する方法には様々なものが知られている。なかでも、キャリルックアヘッド方式加算器 (Carry Look-Ahead Adder), キャリスキップ方式加算器 (Carry-Skip Adder), 及び、キャリセレクト方式加算器 (Carry-select Adder) がよく知られている。これらの方式は、いずれも加算器を経由するキャリの遅れが大きいことに着目し、これを高速化するアイデアを採用している。このうち「キャリセレクト方式加算器」は、従来あまり採用されていなかった。この方式では、回路量が増大してLSIのコストが高くなり、多ビットの加算器に向いていなかったから

である。

【0006】

従来の4ビットのキャリセレクト方式加算器は、仮想キャリ VC_0 を入力とする4ビット加算器と、仮想キャリ VC_1 を入力とする4ビット加算器と、これら2つの4ビット加算器のそれぞれの加算値 S_1 と S_2 とを選択する第1のマルチプレクサと、これら2つの4ビット加算器のそれぞれのキャリ出力 C_1 と C_2 とを選択する第2のマルチプレクサとで構成される。

【0007】

2つの4ビット加算器の構成については特に制約はなく、上記キャリルックアヘッド方式加算器等を採用することができる。仮想キャリ VC_0 は‘0’，仮想キャリ VC_1 は‘1’と定めることができるが、その逆であっても良い。

【0008】

この従来のキャリセレクト方式加算器では、真のキャリ入力 C_{in} を第1，第2のマルチプレクサの選択信号とし、真の加算値 S ，若しくはキャリ出力 C_{out} を選択する。2つの4ビット加算器は、いずれも4ビット入力 P 及び Q を共通にするので、必ず加算値 S_1 若しくは S_2 のいずれかが真の加算値 S になり、対応するキャリ出力 C_1 若しくは C_2 のいずれかが真のキャリ出力 C_{out} になる。

【0009】

このような構成の従来のキャリセレクト方式加算器は、計算速度の面では大きな利点を有している。予め入力 P 及び Q が確定していると、この加算器を継続接続した場合には、ほぼ同時に計算が終了する。このため、その後は真のキャリを選択するだけで済み、加算に要する計算時間を節約できる。即ち、キャリの伝搬を考慮しなくてもよい。

【0010】

【発明が解決しようとする課題】

しかし、従来のキャリセレクト方式加算器は、単なる4ビット加算器と比べると2倍以上の回路量を必要とした。

【0011】

従来のキャリセレクト方式加算器で使用される2つの全加算器 (Full Adder) は、同一の入力値 P_k 及び Q_k を入力し、それぞれ加算値 S_1 または S_2 及びキャリ C_{out1} または C_{out2} を出力する。加算の際のキャリ入力 は、 C_{in1} または C_{in2} であって、上記キャリ出力 C_{out1} または C_{out2} に対応して互いに独立したキャリの経路を作る。そして、マルチプレクサにより、上記加算値 S_1 または S_2 のいずれかをキャリ選択信号 C_s に従って選択し、加算値 S_k として出力する。このような従来のキャリセレクト方式加算器の回路は、全加算器の加算後に、上記マルチプレクサでそれらの加算値を選択するので、回路の無駄が多かった。

【 0 0 1 2 】

そこで、本発明の第1の目的は、従来のキャリセレクト方式加算器を改良して、必要とする回路量を減らし、多ビット乗算器や暗号技術などにおける多ビットの加算に最適な回路構成を提供することにある。また、本発明の第2の目的は、従来の単純なキャリセレクト方式加算器を改良してより高速な加算器を提供することにある。

【 0 0 1 3 】

【課題を解決するための手段】

本発明の第1の目的を達成するために、本発明では加算値の選択をそれぞれの加算器で加算した後に行うのではなく、先に選択したキャリを使って加算する方式を採用し、併せて共通する回路を削減する。

【 0 0 1 4 】

また、本発明の第2の目的を達成するために、本発明では「適応キャリセレクト方式 (Adaptive Carry Select Adder)」と呼ぶ新しい方式を採用し、多ビット値の高速加算を達成する。

【 0 0 1 5 】

即ち、上記目的を達成するための本発明の加算器は、第1と第2の2つのデータ入力と、第1と第2の2つのキャリ入力と、キャリ選択入力とを入力とし、第1と第2の2つのデータ入力のXOR出力を生成する第1のXOR素子と、このXOR出力を選択信号として第1のキャリ入力と第1のデータ入力とのいずれか

を選択する第1のマルチプレクサと、第2のキャリ入力と第2のデータ入力とのいずれかを選択する第2のマルチプレクサと、キャリ選択入力を選択信号として第1と第2の2つのキャリ入力のいずれかを選択する第3のマルチプレクサと、第3のマルチプレクサの出力とXOR出力とのXOR出力を生成する第2のXOR素子とを有する。そして、第1のマルチプレクサの出力を第1のキャリ出力とし、第2のマルチプレクサの出力を第2のキャリ出力とし、第3のマルチプレクサの出力を加算値としたことを特徴とする。

【0016】

このような構成により、従来のキャリセレクト方式加算器よりも必要とする回路量（ゲート数）が少なく、しかも演算速度が高速なキャリセレクト方式の加算器が得られる。

【0017】

また、上記目的を達成するための本発明の他の加算器は、上記構成の加算器が複数段継続接続され、前段の第1のキャリ出力を後段の第1のキャリ入力とし、前段の第2のキャリ出力を後段の第2のキャリ入力とし、複数の加算器のキャリ選択入力を全段で共通とし、初段の加算器の真のキャリ入力をキャリ選択入力とし、第1のキャリ入力を第1の仮想キャリとし、第2のキャリ入力を第2の仮想キャリとしている。そして、キャリ選択入力により最終段の加算器の第1と第2の2つのキャリ出力のいずれかを選択する第4のマルチプレクサを有し、第4のマルチプレクサの出力をキャリ出力としたことを特徴とする。

【0018】

上記構成では、初段の加算器の第3のマルチプレクサを省き、初段の加算器の第3のマルチプレクサの出力の代わりにキャリ選択入力を用いることができる。

【0019】

また、上記目的を達成するための本発明の他の加算器は、上記いずれかの構成の加算器を複数個継続接続し、前段のキャリ出力を後段のキャリ入力とする構成の加算器であって、最終段の加算器を除き、各段の加算器の処理可能なビット数は、その1つ前段の加算器の処理可能なビット数と等しいか、または大きくしたことを特徴とする。

【 0 0 2 0 】

上記構成の加算器では、初段の加算器を除き、各段の加算器の処理可能なビット数と、その1つ前段の加算器の処理可能なビット数との差を一定にすることが望ましい。

【 0 0 2 1 】

また、上記目的を達成するための本発明のブース乗算器は、ツリー構造で構成されるブース乗算器であって、最終段の加算器に、上記複数段の構成の加算器のうちのいずれかを用いたことを特徴とする。

【 0 0 2 2 】

【発明の実施の形態】

以下に、本発明のいくつかの実施の形態を、図面を用いて説明する。なお、本明細書及び図面において、実質的に同一の機能構成を有する構成要素については、同一の符号を付することにより重複説明を省略する。

【 0 0 2 3 】

図1は、本発明の第1の実施の形態によるキャリセレクト方式加算器の1ビット分の構成を示すブロック回路図である。以下、図1に示す構成単位を、R C S A (R e d u c e d C a r r y S e l e c t A d d e r) という。

【 0 0 2 4 】

本発明の第1の実施の形態によるキャリセレクト方式加算器 (R C S A) (1 0 0) は、第1と第2の2つのデータ入力 (P_k , Q_k) と、第1と第2の2つのキャリ入力 (C_{in1} , C_{in2}) と、キャリ選択入力 (C_s) とを入力とする。

【 0 0 2 5 】

そして、第1と第2の2つのデータ入力 (P_k , Q_k) のXOR出力 ($P_k @ Q_k$) を生成する第1のXOR素子 (XOR_1) (1 0 2) と、このXOR出力 ($P_k @ Q_k$) を選択信号として第1のキャリ入力 (C_{in1}) と第1のデータ入力 (P_k) とのいずれかを選択する第1のマルチプレクサ (MPX_1) (1 0 4) と、第2のキャリ入力 (C_{in2}) と第2のデータ入力 (Q_k) とのいずれかを選択する第2のマルチプレクサ (MPX_2) (1 0 5) と、キャリ選択入力

(C_s) を選択信号として第 1 と第 2 の 2 つのキャリ入力 (C_{in1} , C_{in2}) のいずれかを選択する第 3 のマルチプレクサ (MPX_3) (101) と、第 3 のマルチプレクサ (MPX_3) (101) の出力 (C_k) と XOR 出力 ($P_k @ Q_k$) との XOR 出力を生成する第 2 の XOR 素子 (XOR_2) (103) とを有する。

【0026】

そして、第 1 のマルチプレクサ (MPX_1) (104) の出力を第 1 のキャリ出力 (C_{out1}) とし、第 2 のマルチプレクサ (MPX_2) (105) の出力を第 2 のキャリ出力 (C_{out2}) とし、第 3 のマルチプレクサ (MPX_3) (101) の出力を加算値 (S_k) とする (図 1)。なお、ここで記号「@」は、XOR (eXclusive OR) 論理演算子である。

【0027】

図 1 に示す本発明の第 1 の実施の形態によるキャリセレクト方式加算器 RCA (100) では、先にキャリを選択し、その後、選択されたキャリを用いて加算値を計算する方式を採用する。図 1 に示すように、素子 XOR_1 (102) によって出力される、入力 P_k と入力 Q_k との排他論理和 $P_k @ Q_k$ は、第 1 のマルチプレクサ MPX_1 (104) と第 2 のマルチプレクサ MPX_2 (105) のキャリ選択信号として機能する。他方、第 3 のマルチプレクサ MPX_3 (101) は、キャリ入力 C_{in1} または C_{in2} のいずれかを選択し、その選択には、キャリ選択信号 C_s が用いられる。

【0028】

選択されたキャリ C_k は、上記出力 $P_k @ Q_k$ と共に素子 XOR_2 (103) への入力となり、当該ビットにおける加算値 S_k を出力する。また、第 1 のマルチプレクサ MPX_1 (104) は、キャリ入力 C_{in1} または入力 Q_k のいずれかを選択してキャリ C_{out1} として出力する。そして、第 2 のマルチプレクサ MPX_2 (105) は、キャリ入力 C_{in2} または入力 Q_k のうちのいずれかを選択してキャリ C_{out2} として出力する。

【0029】

論理記号で考察した場合、通常、加算値 S とキャリ C_{out} との関係は、

$$S = P @ Q @ C_{in} \quad \dots \text{式1}$$

$$C_{out} = PQ + QC_{in} + PC_{in} \quad \dots \text{式2}$$

と表現される。しかし、 C_{out} については上記の式2ではなく、次式の、

$$C_{out} = C_{in} (P @ Q) + P (P @ Q) _ \quad \dots \text{式3}$$

若しくは、

$$C_{out} = C_{in} (P @ Q) + Q (P @ Q) _ \quad \dots \text{式4}$$

を使用する方がよい。但し、ここで記号「 $_$ 」は否定論理である。

【0030】

その第1の理由は、式2では、3入力ANDOR論理を取る必要があり、回路が複雑化するからである。そして第2の理由は、図1のマルチプレクサ MPX_1 (104)と第2のマルチプレクサ MPX_2 (105)のキャリ選択信号として排他論理和 $P_k @ Q_k$ を用いることに関連するが、キャリセレクト方式加算器では、入力PやQはキャリ C_{in} より先に定まるので、マルチプレクサで選択する方式が速度的にも回路量的にも有利だからである。

【0031】

図1の回路を実際にLSIで実現する場合には、トランスミッションゲート (Transmission Gate) を用いれば回路量が削減できることが知られている。トランスミッションゲートを用いる論理 (以下、「TG論理」という。) では、見かけ上ゲート数が少なく済む。しかし、ドライブ能力を欠くことがあるので、設計上注意が必要である。例えば、必要に応じてインバータを追加し、若しくは負論理で動作させるなどの工夫が効果的である。

【0032】

図2は、TG論理によるXOR論理回路であって、図1の素子 XOR_1 (102)で使用することができる。回路的には、2つのインバータと1つのTGから構成されるが、一方のインバータINVは、そのソース側が入力 Q_k 若しくは $Q_k _$ に接続されるのが通常と異なる点である。図2で、基盤に矢印を持つトランジスタがPチャンネルMOSであり、矢印のないものがNチャンネルMOSである。この様にTG論理を使うとゲート数を削減できるのは、トランスミッションゲートとトグルスイッチとの類似性にあると考えられる。図2の回路では、式1

の前半部分、及び、式 3、若しくは、式 4 の選択信号 $P_k @ Q_k$ を計算することができる。

【 0 0 3 3 】

図 3 は、TG 論理による XOR 論理回路であって、図 1 の素子 XOR₂ (1 0 3) で使用することができる。回路的には、2 つのインバータと 2 つの TG から構成される。キャリ選択信号 $P_k @ Q_k$ により、キャリ C_k 若しくは C_k — が選択されるので、結果的に式 1 の加算値 S_k を計算することができる。

【 0 0 3 4 】

図 4 は、TG 論理によるマルチプレクサであって、図 1 のマルチプレクサ MPX₁ (1 0 4)、または、マルチプレクサ MPX₂ (1 0 5) で使用することができる。回路的には、キャリ選択信号 $P_k @ Q_k$ によりキャリ入力 C_{in1} 若しくはキャリ入力 C_{in2} か Q_k かを選択することにより、式 4 によるキャリ出力 C_{out1} 若しくは C_{out2} を計算することができる。入力を Q_k ではなく P_k とすることにより、式 3 へ適用することもできる。

【 0 0 3 5 】

図 5 は、TG 論理によるマルチプレクサであって、図 1 の第 3 のマルチプレクサ MPX₃ (1 0 1) で使用することができる。真のキャリ C_s をキャリ選択信号として用いて、仮想キャリ入力 C_{in1} 若しくは C_{in2} のいずれかを真のキャリ C_k として選択することができる。

【 0 0 3 6 】

本発明の第 1 の実施の形態による図 1 の回路構成の加算器を、従来の回路構成の加算器と比較した場合に、単純に TG 論理で構成したとすときの対照表を表 1 に示す。単なる加算器は 20 ゲートで構成できるので、TG 論理の優秀性は明らかである。但し、配線容量で遅延量が大きく、ドライブ能力が欠けることがあるので設計上は注意が必要である。表 1 に示すように、従来の回路構成による加算器では 46 ゲートを要したが、本発明の第 1 の実施の形態による図 1 の回路構成による加算器では、32 ゲートで済ませることができ、3 割程度の回路量を削減することができた。また、従来の回路構成による 4 ビット加算器では 190 ゲートを要したが、本発明の第 1 の実施の形態の回路構成による 4 ビット加算器では

， 1 3 4 ゲートで済ませることができ，ここでも 3 割程度の回路量を削減することができた。

【 0 0 3 7 】

【表 1】

	T G 論理による回路量 (ゲート数)
単なる加算器	2 0
従来方式	4 6
本発明方式	3 2
従来方式による 4 ビット加算器	1 9 0
本発明方式による 4 ビット加算器	1 3 4

【 0 0 3 8 】

計算速度の面では，キャリ入力からキャリ出力に至るまでにマルチプレクサ 1 段分の遅延しか要しないことは重要である。この事実，後述する本発明の第 2 の実施の形態の基礎を与える。キャリセレクト方式加算器では，入力 P や Q はキャリ C_{in} より先に定まるので，マルチプレクサのキャリ選択信号 P @ Q は，キャリの伝搬に殆ど関与しない。T G 論理によれば，わずか 1 段分のゲート遅延でキャリは 1 つの加算器を伝搬することができる。インバータを追加し，若しくは負論理で動作させたとしても，ゲート遅延 2 段分で十分である。

【 0 0 3 9 】

図 1 の本発明の第 1 の実施の形態によるキャリセレクト方式加算器の 1 ビット分の回路から，多ビットの加算器を容易に構成することができる。図 6 は，本発明の第 1 の実施の形態による R C S A (1 0 0) を採用した，4 ビットキャリセレクト方式加算器の構成図である。この加算器 (2 0 4) は，第 1 の実施の形態による R C S A (2 0 1 ~ 2 0 4) が，4 ビット分継続接続されている。キャリセレクト方式加算器では真のキャリ C_{in} の値は ' 0 ' または ' 1 ' のいずれかであるので，仮想キャリを用いて加算を先に進めておき，最後に真のキャリ入力 C_{in} で真のキャリ出力 C_{out} ，及び加算値 S_0 ， S_2 ，または S_3 を選択す

る。このような構成では、加算値 S_0 , S_2 , または S_3 の選択がそれらの加算後に行われるのではなく、選択されたキャリ C_k を使う加算が行われるところが従来と異なる。これは、加算値の計算はキャリの生成と比べて遅くても良いと言う事情を利用している。

【 0 0 4 0 】

図6において、 $RCSA_0$ (201), $RCSA_1$ (202), $RCSA_2$ (203), 及び $RCSA_3$ (204) は、図1の構成を持つ本発明の実施の形態による簡易キャリセレクト方式加算器であり、 $RCSA_0 \sim RCSA_3$ (201 ~ 204) は、順に継続接続されている。この加算器では、仮想キャリを定めて仮のキャリ出力 C_{out1} 若しくは C_{out2} を計算し、真のキャリ入力 C_{in} であるキャリ選択信号 C_s を用いて、マルチプレクサ MPX (205) において真のキャリ出力 C_{out} を選択する。なお、初段の加算器 $RCSA_0$ (201) において、図1の第3のマルチプレクサ MPX_3 (101) を省略し、直接真のキャリ入力 C_{in} を選択後のキャリ C_k とすることができる。また、仮想キャリ VC_0 は、値' 0' , 仮想キャリ VC_1 は値' 1' と決めておくが、その逆であってもよい。 VC_0 と VC_1 の値を逆にしたときはマルチプレクサ MPX (205) において、上記と逆の選択をすればよい。

【 0 0 4 1 】

図7は、本発明の第1の実施の形態による4ビット加算器を採用した16ビットキャリセレクト方式加算器の一実施形態である。この加算器は図7に示すように、図6の4ビットキャリセレクト方式加算器(4 $RCSA$)を継続接続して、16ビットキャリセレクト方式加算器を構成している。この実施形態では、継続する4 $RCSA$ は3段のみで、初段は通常の4ビット加算器(301)を使用している。この実施形態は4ビット加算器による一種の並列演算であって、初段の加算器はキャリ選択回路が不要なことから回路量を減らす工夫をしている。従って、4ビット加算器(301)の方式は特に制限されず、通常のCPAであっても、キャリルックアヘッド方式加算器、若しくは、キャリスキップ方式加算器であってもよい。

【 0 0 4 2 】

4ビット加算器(301)において、Xbus(305)からの4ビット入力P3:0(0ビットから3ビットまでの入力を表す。以下同様。)と、Ybus(306)からの4ビット入力Q3:0とを、キャリ入力C_{in}を受けて加算し、加算値として4ビット出力S3:0と、キャリを出力する。この出力されたキャリは、次段の加算器4RC SA₁(302)のキャリ入力となる。次段の加算器4RC SA₁(302)は、図6のキャリセレクト方式加算器であって、Xbus(305)からの4ビット入力P7:4と、Ybus(306)からの4ビット入力Q7:4とを、キャリ入力C_iを受けて加算し、加算値として4ビット出力S7:4と、キャリC₀を出力する。このキャリC₀は、3段目の加算器4RC SA₂(303)のキャリ入力となる。

【0043】

3段目の加算器4RC SA₂(303)は、図6のキャリセレクト方式加算器であって、Xbus(305)からの4ビット入力P11:8と、Ybus(306)からの4ビット入力Q11:8とを、キャリ入力C_iを受けて加算し、加算値として4ビット出力S11:8と、キャリC₀を出力する。このキャリC₀は、3段目の加算器4RC SA₂(303)のキャリ入力となる。4段目の加算器4RC SA₃(304)は、図6のキャリセレクト方式加算器であって、Xbus(305)からの4ビット入力P15:12と、Ybus(306)からの4ビット入力Q15:12とを、キャリ入力C_iを受けて加算し、加算値として4ビット出力S15:12と、キャリ出力C_{out}を出力する。

【0044】

図8は、図7に示す本発明の実施形態によるキャリセレクト方式加算器を用いた16ビット加算器の動作を説明するための、出力端子等の波形変化の形で遅延時間を表示したタイムチャートである。図8では、RD__信号によってXbusとYbusのデータが確定し、加算結果が遅延時間Tdの後、Zbus上に現れるまでの変化により、本実施形態によるキャリセレクト方式加算器の高速性が示されている。図8で加算キャリの遅延時間をTaとした場合、4ビット加算器4RC SA₀(301)の出力S3:0(0ビットから3ビットまでの出力を表す。以下同様。)は、遅延時間4Taで加算が終了する。また、選択キャリのマル

チプレクサでの遅延時間を T_s とした場合、加算器 $4RCSA_1$ (302) の出力 $S7:4$ は、遅延時間 $4Ta + Ts$ で加算が終了する。また、加算器 $4RCSA_2$ (303) の出力 $S11:8$ は、遅延時間 $4Ta + 2Ts$ で加算が終了する。また、加算器 $4RCSA_3$ (304) の出力 $S15:12$ は、遅延時間 $4Ta + 3Ts$ で加算が終了する。従って、加算結果が $Zbus$ 上に現れるまでの遅延時間 Td は、最大で $4Ta + 3Ts$ 程度である。なお、通常の 16 ビット加算器の出力の加算時間は、 $16Ta$ である。

【0045】

本実施形態のポイントは、データが確定した後に、4つの加算器 (301~304) が仮想キャリによって同時に加算を始めることができる点にある。このようなことが可能になった結果、4ビット分の加算に要する時間はとられてしまうが、その後は選択キャリがマルチプレクサを通過する遅延時間 T_s だけで桁数の多い加算を実行することができるという利点を享受できる。本実施形態では、キャリは加算器上ではなく、マルチプレクサ上を伝搬して行くと捉えることもできる。

【0046】

以上示したように、本発明の第1の実施の形態によれば、 $RCSA$ の簡易なキャリセレクト方式加算器によって、従来のキャリセレクト方式加算器と比べて3割程度回路量を削減することができる。計算速度の面では、キャリ入力からキャリ出力に至るのにマルチプレクサ1段分の遅延しか要しない。

【0047】

(第2の実施の形態)

本発明による第2の実施の形態は、本明細書で適応キャリセレクト方式 (Adaptive Carry-select Adder) と呼ぶ新しい方式に関する。この方式は、キャリ選択信号 Cs を用いるマルチプレクサ (例えば、図6の MPX) の遅延時間 T_s と、本実施形態によるキャリセレクト方式加算器のキャリ遅延時間 T_a とが、ほぼ同等である事実を基礎としている。これは加算器を通過するキャリ (以下、加算キャリという。) が、図1の第1のマルチプレクサ MPX_1 または第2のマルチプレクサ MPX_2 の1段分の遅延 (T_a) を有し

、キャリ選択信号 C_s が、図 6 のマルチプレクサ M P X の 1 段分の遅延 (T_s) を有し、 T_a と T_s がほぼ同等である。このことから、多ビットの加算において、次段のキャリセレクト方式加算器のビット数を増やしても、キャリ生成の遅延時間は変わらないことを利用している。この遅延時間は変わらないが、処理できるビット数は漸次増やしていけるので、全体の加算時間は短縮することになる。

【 0 0 4 8 】

図 9 は、本発明の第 2 の実施の形態による多ビット加算器の回路構成である。この加算器は、上述の適応キャリセレクト方式で構成したことを特徴とする。この加算器は、第 1 の実施の形態による加算器 ($A_0 \sim A_n$) が複数段継続接続されている。前段の第 1 のキャリ出力を後段の第 1 のキャリ入力とし、前段の第 2 のキャリ出力を後段の第 2 のキャリ入力とし、複数の加算器のキャリ選択入力を全段で共通とする。そして、初段の加算器の真のキャリ入力をキャリ選択入力とし、第 1 のキャリ入力を第 1 の仮想キャリとし、第 2 のキャリ入力を第 2 の仮想キャリとし、キャリ選択入力により最終段の加算器の第 1 と第 2 の 2 つのキャリ出力のいずれかを選択する第 4 のマルチプレクサを有する。そして、第 4 のマルチプレクサの出力をキャリ出力としている。なお、初段の加算器の第 3 のマルチプレクサは省かれ、初段の加算器の第 3 のマルチプレクサの出力の代わりにキャリ選択入力を用いている。

【 0 0 4 9 】

図 9 で、初段の M_0 ビット加算器 A_0 は、第 1 の実施の形態による R C S A である。そして、次段の M_1 ビット加算器 $A_1 \sim$ 最終段の M_n ビット加算器 A_n も、全て第 1 の実施の形態による R C S A で構成されている。

【 0 0 5 0 】

従って、 N ビット加算器を対象とした場合、

$$N = \sum M_k \quad (k = 0, \dots, n) \quad \dots \quad \text{式 5}$$

が成立する。但し、最終段加算器 A_n のビット数 M_n は、必ずしも前段のビット数より大きいとは限らない。数 N が丁度良い値でなければ、通常は M_n は端数になるからである。

【 0 0 5 1 】

ここで、 $M_1 = M_0$ のように構成すると、最初のマルチプレクサに加算器 A_0 と A_1 のキャリをほぼ同時に入力することができる。これは入力側での最適化を意味する。また、最終段加算器の 1 つ手前の加算器 A_{n-1} において、その出力側のマルチプレクサに加算器を経由したキャリとキャリセレクト信号が同時に入力することができれば、全体の加算時間 (T_t) を最適化することができる。これは出力側での最適化を意味し、上記入力側での最適化と区別できる。最終段加算器 A_n を対象にしなかったのは、そのビット数 M_n が通常端数になるからである。

【 0 0 5 2 】

上記選択キャリ、即ち、加算器 A_{n-1} に至るキャリセレクト信号 C_s の遅れ T_d は、

$$T_d = M_0 T_a + (n-2) T_s \quad \dots \quad \text{式 6}$$

で与えられる。他方、加算器 A_{n-1} を経由した前記加算キャリの遅れが同じく T_d であれば、出力側での最適化を達成することができ、

$$T_d = M_{n-1} \cdot T_a \quad \dots \quad \text{式 7}$$

で与えられる。なお、全体の加算時間 T_t は、

$$T_t = T_d + 2 T_s \quad \dots \quad \text{式 8}$$

である。

【 0 0 5 3 】

第 2 の実施の形態において、漸次増やした加算器のビット数を等差数列で構成してみる。すると、公差を h ビットとした場合、

$$M_k = M_0 + h (k-1) \quad (k=1, \dots, n-1) \quad \dots \quad \text{式 9}$$

と表現することができる。

【 0 0 5 4 】

式 6 及び式 7 より公差 h を求めると、

$$h = T_s / T_a \quad \dots \quad \text{式 10}$$

となる。そして、式 5 及び式 9 より n を求めると、

$$n = \{ - (2 M_0 - 3 h) + \sqrt{ (2 M_0 - 3 h)^2 + 8 h (N - M_n - h) } \} / 2 h \quad \dots \quad \text{式 11}$$

となる。

【 0 0 5 5 】

また、全体の加算時間 T_t を最小化するために初期値 M_0 をどのように定めるのが良いかを計算すると、

$$M_0 = 3h / 2 \quad \dots \quad \text{式 1 2}$$

となり、このときキャリを選択するマルチプレクサの個数 n は、

$$n = \sqrt{\{2(N - M_n - h) / h\}} \quad \dots \quad \text{式 1 3}$$

と求まる。そして、 T_t の最小値 $(T_t)_{\min}$ は、式 1 3 の n を用いて、

$$(T_t)_{\min} = (n + 3 / 2) h \cdot T_a \quad \dots \quad \text{式 1 4}$$

で与えられる。

【 0 0 5 6 】

公差 h は式 1 0 で与えられるので、上記考察に基づき 1 に近い値であると推定できる。第 1 の実施の形態による RC SA を採用した場合には、選択キャリのファンアウトが多いので、加算キャリの遅延時間 T_a よりも選択キャリの遅延時間 T_s の方が大きくなることがある。従って、公差 h は 1 に近く、かつ、1 より大きい数になることがある。しかし、第 1 の実施の形態による RC SA を採用しない場合は、加算キャリが選択キャリよりも遅いのが通常で、従って h は通常 1 よりも小さくなる。

【 0 0 5 7 】

実際の回路では必ずしも全体の加算時間 T_t を最小化するように上記変数を定めることはできない。変数 h は、1 とか、2 とか、0.5 とか、整数若しくは整数分の 1 を採用しなければならず、また、初段の加算器 A_0 のビット数 M_0 も整数でなければならない。従って、上記最適化に沿った最善の設計が必要になる。

【 0 0 5 8 】

図 1 0、図 1 1 は、上記観点から、多ビット加算器での本実施形態による具体的構成を例示したものである。図 1 0 (a) は、64 ビット加算器及び 128 ビット加算器について従来の構成法を例示してある。ここでは単純に 4 ビット加算器を継続接続し、その都度真のキャリを選択する。この方式では、選択キャリの遅延時間が単純に加算されていくので無駄が多い。

【 0 0 5 9 】

図 1 0 (b) は、本発明の第 2 の実施の形態であって、初期値 $M_0 = 1$ で、公差 $h = 1$ の場合である。初期値 $M_0 = 1$ であることから、初段の加算器 A_0 は 1 ビット、次段の加算器 A_1 も 1 ビットである。以下、公差 1 で加算器のビット数が増加する構成を採用する。64 ビット加算器の場合 ($N = 64$)、 $n = 10.7$ であるので、 A_{10} は 10 ビット加算器で、最後の加算器 A_{11} は端数ビット、 $M_n = 8$ になる。64 ビット加算器の場合 ($N = 128$)、 $n = 15.4$ であるので、 A_{15} は 15 ビット加算器で、最後の加算器 A_{16} は端数ビット、 $M_n = 7$ になる。 M_n を最初から定めることができないので、実際には式 11 から n を求めることはできず、およその n の値しか知ることはできない。実際に M_n を定めるには、 n を求めた後、式 5 により再計算することになる。

【 0 0 6 0 】

図 1 0 (c) は、本発明の第 2 の実施の形態であって、初期値 $M_0 = 1$ で、公差 $h = 0.5$ の場合である。公差 $h = 0.5$ を実現するために加算器のビット数を加算器 2 個毎に 1 ビットずつ増加させている。公差が $1/2$ の場合は、3 番目の加算器 A_2 の出力側のマルチプレクサにおいて、選択キャリよりも加算器 A_2 の加算キャリの方が遅くなり、この事情は後段の加算器でも同様である。そこで、全体の計算時間 T_t を求めるに際し、端数 M_n が出る場合は加算器 A_{n-1} の遅延時間に基づいて計算し、その値は $M_{n-1} \cdot T_a + 2 T_s$ である。幸いにし、キャリセレクト方式加算器においては遅延時間の累積は起こらない。即ち、一種の並列計算であるので、前段の計算で生じた遅延が後段の計算時間を積み増すことはない。

【 0 0 6 1 】

図 1 1 (d) は、本発明の第 2 の実施の形態であって、初期値 $M_0 = 1$ で、公差 $h = 2$ の場合である。公差 $h = 2$ 、即ち選択キャリの方が加算キャリよりも倍も遅いという事態は、本実施の形態の構成を採用する限り、余り起こらない。但し、キャリ選択信号 C_S のファンアウトが大きく、長い配線が想定されるので、全く起こらないとも言い切れない。ここでは比較のために例示した。

【 0 0 6 2 】

図 1 1 (e) は、本発明の第 2 の実施の形態であって、初期値 $M_0 = 2$ で、公差 $h = 1$ の場合である。上述の如く、全加算時間 T_t を最小化するための初期値 M_0 の値は式 1 2 で与えられ、公差 $h = 1$ の場合 $M_0 = 1.5$ とするのが最適である。しかし、 M_0 は整数を選ばなければならないので、初期値 $M_0 = 2$ とする設計を採用することもできる。図 1 0 (b) の $M_0 = 1$ の場合と比べると、全加算時間 T_t は殆ど同じである。図 1 1 (f) や図 1 1 (g) でも同様のことが言える。表 2 では、全加算時間 T_t に関し、例えば $h = 1$ の場合、理論上の最適値と最善の設計の値との差は 1 % 以下であり、この様な設計の有効性を示している。

【 0 0 6 3 】

表 2 に、図 1 0、図 1 1 で掲げた本発明の第 2 の実施の形態について、全加算時間 T_t を計算した値を列挙した。図 1 0 (a) の従来例と比べると、64 ビット加算器で 4 割程度、128 ビット加算器で 5 割程度、全加算時間 T_t を減らすことができる。勿論通常の C P A と比べると格段に速い。全加算時間 T_t は最終段の 1 つ手前の加算器 A_{n-1} の加算キャリの遅延時間程度であり、より正確には、 $h = 1$ の場合 C P A の全加算時間との比はおおよそ、 $N / (M_{n-1} - 2)$

である。この値は 64 ビット加算器で 8 倍、128 ビット加算器で 10 倍程度である。

【 0 0 6 4 】

【表 2】

加算器の構成方法	M_0	h	全加算時間 T_t (T_t/T_a)		備考
			64ビット	128ビット	
図10 (a)	4	0	$4T_a + 15T_s$ —	$4T_a + 31T_s$ —	従来例
図10 (c)	1	1/2	$8T_a + T_s$ (8.5)	$11T_a + 2T_s$ (12)	本発明例
図10 (b)	1	1	$T_a + 11T_s$ (12)	$T_a + 16T_s$ (17)	〃
図11 (d)	1	2	$T_a + 8T_s$ (17)	$T_a + 12T_s$ (25)	〃
図11 (f)	2	1/2	$8T_a + 2T_s$ (9)	$11T_a + 2T_s$ (12)	〃
図11 (e)	2	1	$2T_a + 10T_s$ (12)	$2T_a + 15T_s$ (17)	〃
図11 (g)	2	2	$2T_a + 8T_s$ (18)	$2T_a + 11T_s$ (24)	〃
理論上の最適値	3/2 h	h	$1.5T_a + 10.5T_s$ (12.0)	$1.5T_a + 15.5T_s$ (17.0)	h = 1 と仮定

【0065】

表2において、公差 h が $1/2$ の場合は、3番目の加算器 A_2 の出力側のマルチプレクサにおいて、選択キャリよりも加算器 A_2 の加算キャリの方が遅くなる。この事情は後段の加算器でも同様である。

【0066】

回路量でみた場合、図10 (a) の従来例と比べると、加算器の総ビット数 N は同じであるが、キャリの選択に要するマルチプレクサの数を減らせることができる。具体的には、64ビット加算器で4個程度、128ビット加算器で15個程度減らせる。

【0067】

以上示したように、本発明の第2の実施の形態によれば、本明細書で適応キャリセレクト方式と呼ぶ新しい方式を採用し、多ビット値の高速加算が達成され、併せて回路量の削減をすることができる。

【 0 0 6 8 】

(第 3 の実施の形態)

図 1 2 は、本発明の第 2 の実施形態の適応キャリセレクト方式を、32ビットブース乗算器 (Booth Multiplier) に応用した第 3 の実施の形態の構成を示すブロック回路図である。図 1 2 では、2 次ブース乗算器のワレスツリー (Wallace Tree) を、CSA (Carry Save Adder) で構成している。P₁₆ ~ P₀ の 17 個の値は、2 次ブース乗算器の部分積であって、それぞれ 33 ビットで構成される。3-2 CSA (148) は、3 つの数値を 2 つにまとめることができる CSA であり、これによって加算に要する数値の削減を行う。また、4-2 CSA (141, 142, 143, 144, 145, 146, 147) は、4 つの数値を 2 つにまとめることができる CSA であり、3-2 CSA を継続接続して構成する。

【 0 0 6 9 】

通常の 2 次ブース乗算器のワレスツリーでは、最終段加算器の位置には ACSSA (149) ではなく、CPA (Carry Propagation Adder) が用いられ、2 つの数値の加算が行われる。この CPA は最終的に残った 2 つの 64 ビット値を加算するためのもので、65 ビット構成であり、唯一キャリの伝搬で大きな遅延が発生し、乗算器の高速化のネックとなっていた部分である。第 3 の実施形態は、適応キャリセレクト方式加算器 (ACSSA) をこの CPA の代わりに採用したもので、32 ビット乗算器の高速化に著しい寄与をなしている。なお、2 の補数表示により 32 ビット値の先頭ビット (MSB) を符号にした場合も同様である。

【 0 0 7 0 】

将来の 64 ビット乗算器等を視野に入れた場合に、多ビット値の高速加算は必須の技術に成りつつあり、本発明の第 3 の実施の形態による ACSSA を採用した高速乗算器は、単なる構成要素の置き換えに留まらない効果を発揮する。例えば、64 ビット乗算器を 2 次のブース乗算器で構成した場合、上記 ACSSA (149) は、32 ビット乗算器の場合のおよそ 2 倍の多ビット値を取り扱うことになる。これを従来通りの CPA で構成した場合、そのキャリの遅延時間は乗算器の

高速化の大きな障害となるであろうし、64ビット乗算器そのものの採用意義を失わせることにもなりかねない。

【0071】

以上示したように、本発明の第3の実施の形態によれば、2次ブース乗算器のワレスツリーへの応用により、乗算器の高速化に著しく寄与することができる。

【0072】

なお、上記説明した実施形態の加算器、乗算器は、単独ではもちろん、集積回路の一部として組み込まれてもその効果を発揮することはいうまでもない。

【0073】

以上、添付図面を参照しながら本発明の加算器、乗算器、及びこれを用いた集積回路の好適な実施形態について説明したが、本発明はこれらの例に限定されない。いわゆる当業者であれば、特許請求の範囲に記載された技術的思想の範疇内において各種の変更例または修正例に想到し得ることは明らかであり、それらについても当然に本発明の技術的範囲に属するものと了解される。

【0074】

【発明の効果】

本発明により、従来のキャリセレクト方式加算器を改良して、多ビット乗算器や暗号技術などにおける多ビットの加算に最適な、回路量を減らし、より高速とした加算器、乗算器の回路構成が提供できた。

【図面の簡単な説明】

【図1】

図1は、本発明の第1の実施の形態によるキャリセレクト方式加算器の1ビット分の構成を示すブロック回路図である。

【図2】

図2は、TG論理によるXOR論理回路である。

【図3】

図3は、TG論理によるXOR論理回路である。

【図4】

図4は、TG論理によるマルチプレクサである。

【図 5】

図 5 は、T G 論理によるマルチプレクサである。

【図 6】

図 6 は、本発明の第 1 の実施の形態による R C S A を採用した 4 ビットキャリセレクト方式加算器の構成図である。

【図 7】

図 7 は、本発明の第 1 の実施形態による 4 ビット加算器を採用した 1 6 ビットキャリセレクト方式加算器の回路構成図である。

【図 8】

図 8 は、図 7 に示すキャリセレクト方式加算器を用いた 1 6 ビット加算器の動作を説明するための、出力端子等の波形変化の形で遅延時間を表示したタイムチャートである。

【図 9】

図 9 は、本発明の第 2 の実施の形態による多ビット加算器の回路構成図である。

【図 1 0】

図 1 0 は、(a) 従来構成と、(b) , (c) 本発明の第 2 の実施の形態による多ビット加算器の具体的構成を例示した回路図である。

【図 1 1】

図 1 1 (d) ~ (g) は、本発明の第 2 の実施の形態による多ビット加算器の具体的構成を例示した回路図である。

【図 1 2】

図 1 2 は、本発明の第 3 の実施の形態による 3 2 ビットブース乗算器の一例を示す構成図である。

【符号の説明】

1 0 0 R C S A

P_k 第 1 のデータ入力

Q_k 第 2 のデータ入力

$C_{i n 1}$ 第 1 のキャリ入力

C_{in2} 第2のキャリ入力

C_s キャリ選択入力

C_{out} キャリ出力

101 第3のマルチプレクサ (MPX_3)

102 第1のXOR素子 (XOR_1)

103 第2のXOR素子 (XOR_2)

104 第1のマルチプレクサ (MPX_1)

105 第2のマルチプレクサ (MPX_2)

201 $RCSA_0$

202 $RCSA_1$

203 $RCSA_2$

204 $RCSA_3$

205 マルチプレクサ

S 加算値

C_{out1} , C_{out2} 仮のキャリ出力

301 通常の4ビット加算器

302 加算器4 $RCSA_1$

303 加算器4 $RCSA_2$

304 加算器4 $RCSA_3$

305 $Xbus$

306 $Ybus$

307 $Zbus$

C_0 キャリ

$A_0 \sim A_n$ 加算器

400 4ビットキャリセレクト方式加算器

401, 402 4ビット加算器

403, 404 マルチプレクサ

500 1ビット分の回路

501, 502 全加算器

5 0 3 マルチプレクサ

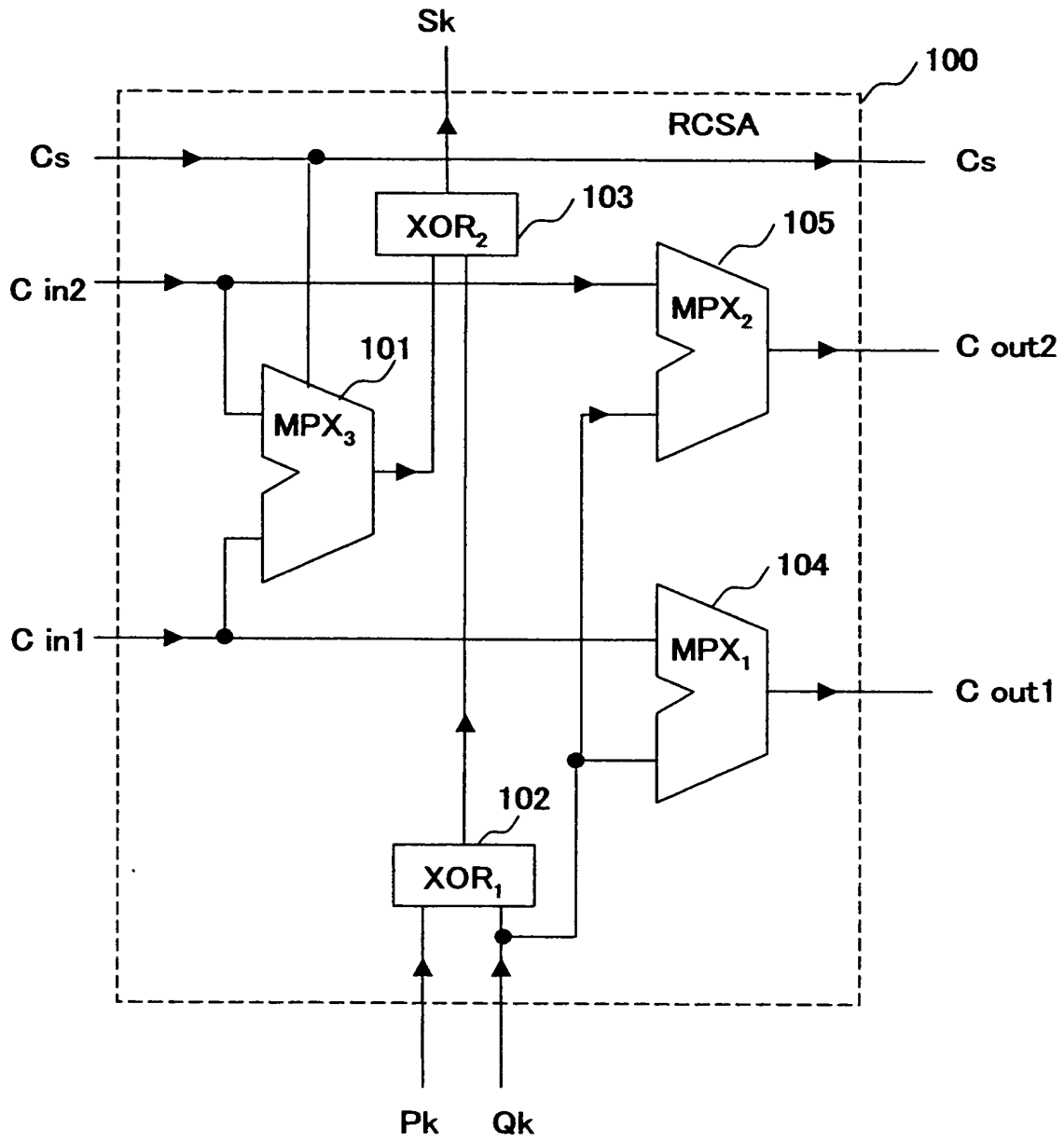
1 4 1 ~ 1 4 7 4 - 2 C S A

1 4 8 3 - 2 C S A

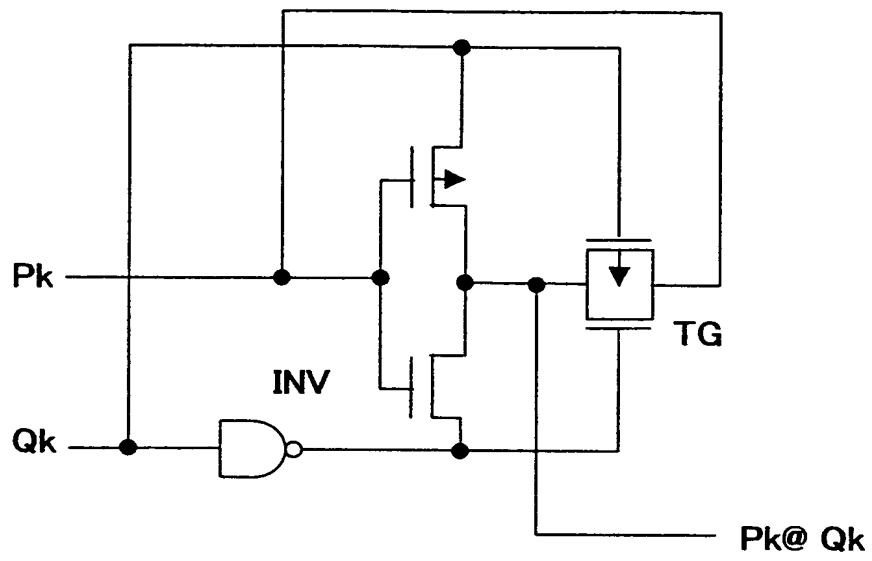
1 4 9 A C S A

【書類名】 図面

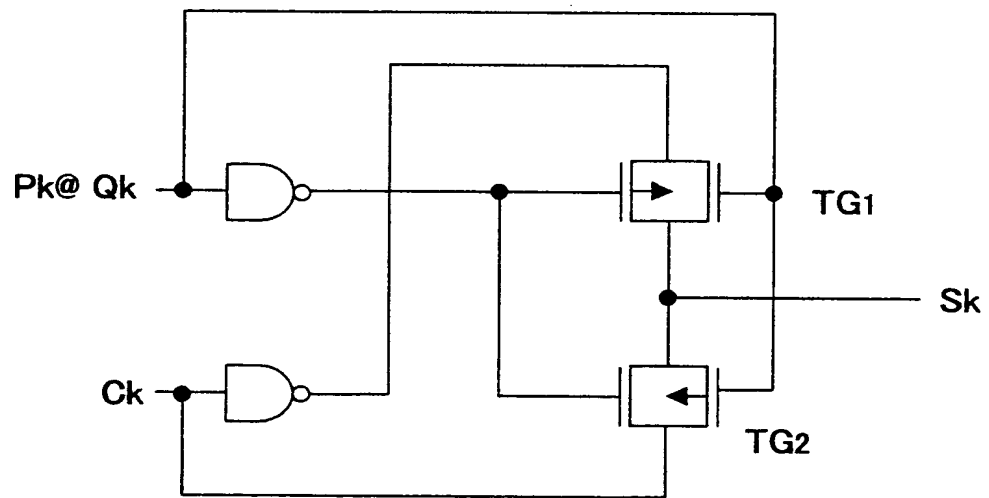
【図 1】



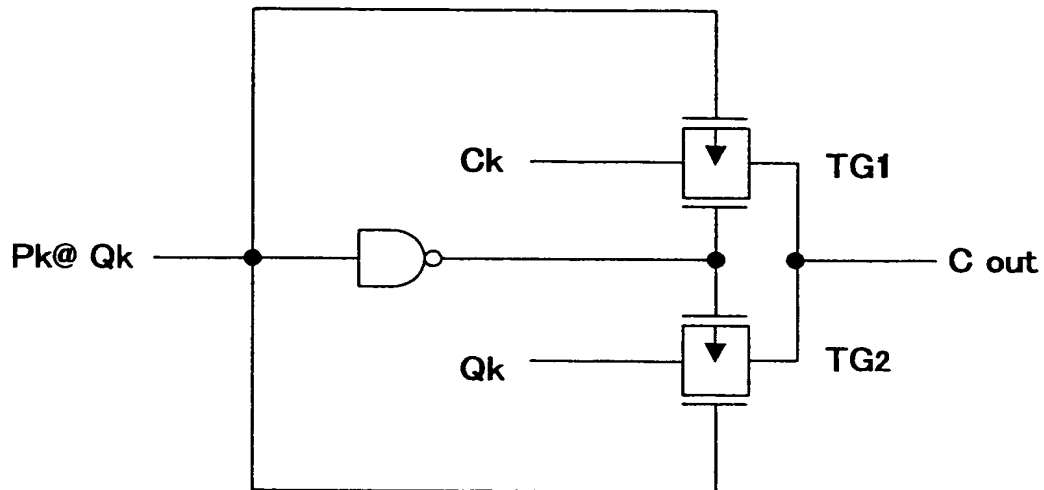
【図 2】



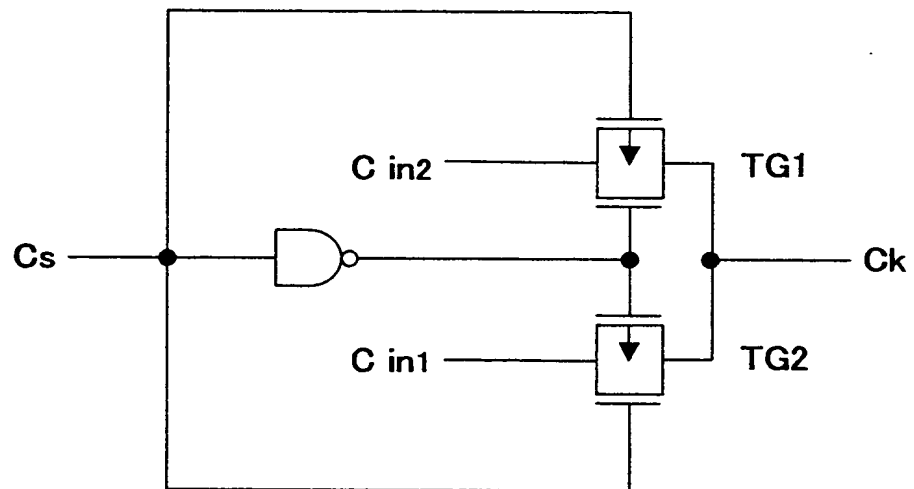
【図 3】



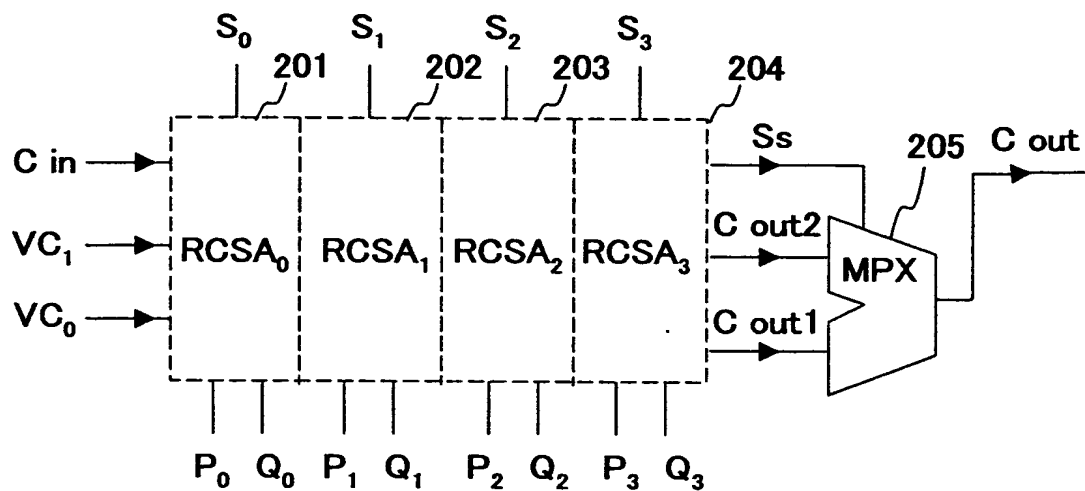
【図 4】



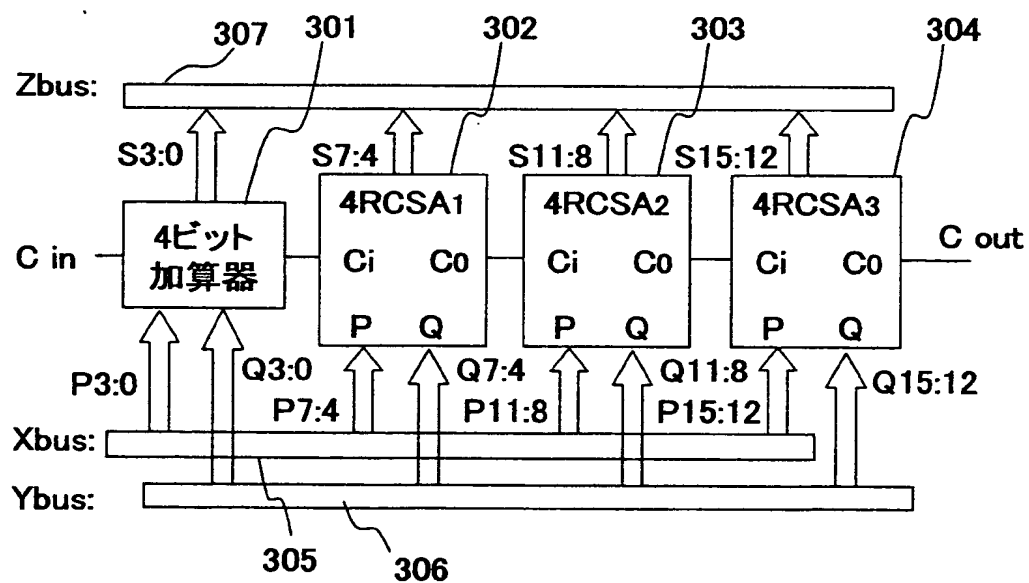
【図 5】



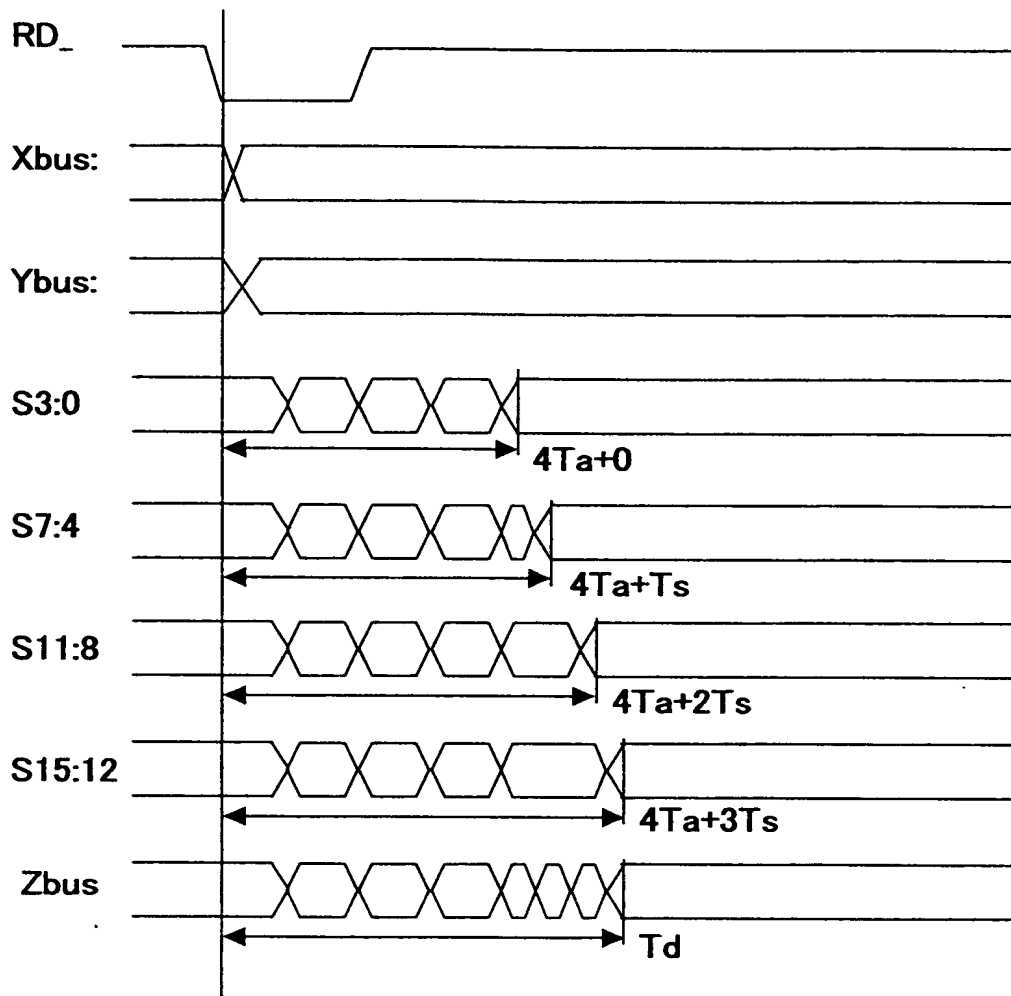
【図 6】



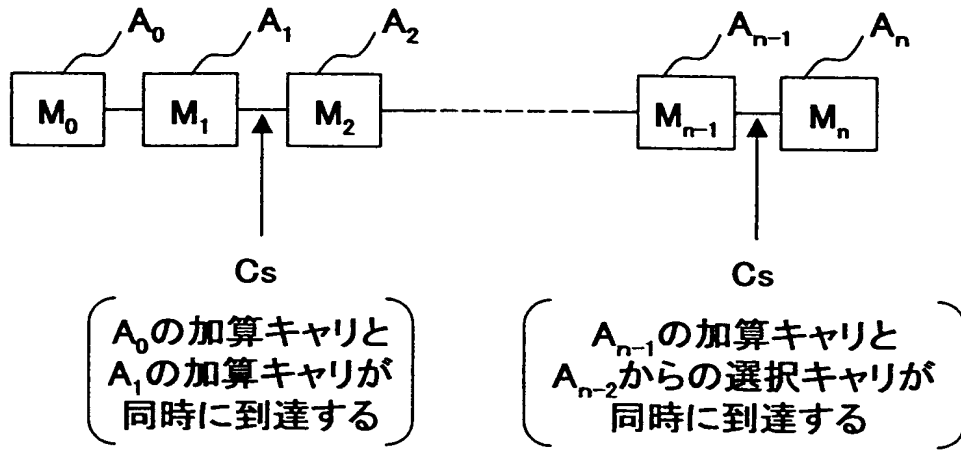
【図 7】



【図 8】



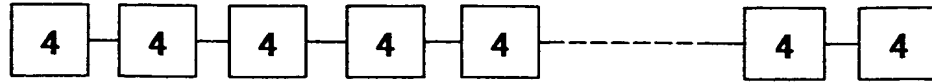
【図 9】



【図 1 0】

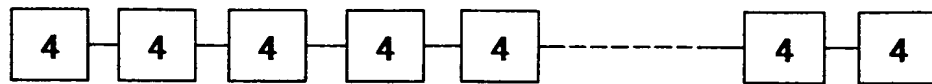
(a)従来の構成

64ビット加算器の場合



4RCSAを16個継続接続する。

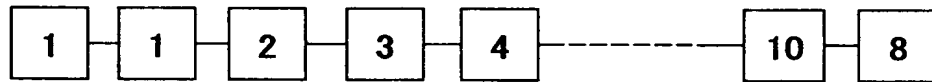
128ビット加算器の場合



4RCSAを32個継続接続する。

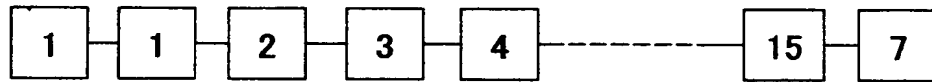
(b)初期値 $M_0=1$ 、公差 $h=1$ の場合

64ビット加算器の場合



RCSAのビット数が漸次増加する。

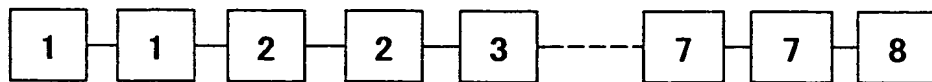
128ビット加算器の場合



RCSAのビット数が漸次増加する。

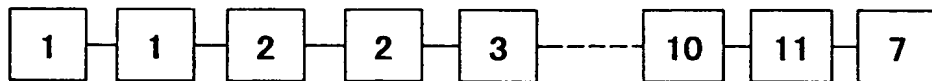
(c)初期値 $M_0=1$ 、公差 $h=0.5$ の場合

64ビット加算器の場合



RCSAのビット数が2個おきに漸次増加する。

128ビット加算器の場合

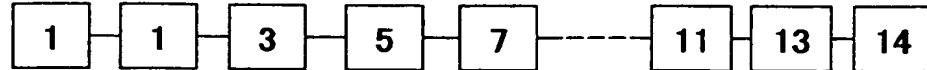


RCSAのビット数が2個おきに漸次増加する。

【図 1 1】

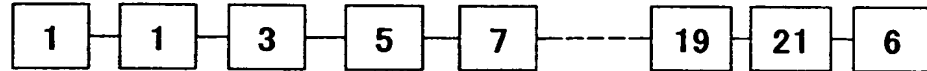
(d)初期値 $M_0=1$ 、公差 $h=2$ の場合

64ビット加算器の場合



RCSAのビット数が2個ずつ漸次増加する。

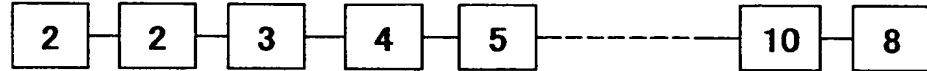
128ビット加算器の場合



RCSAのビット数が2個ずつ漸次増加する。

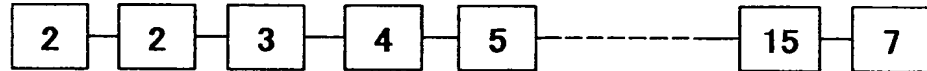
(e)初期値 $M_0=2$ 、公差 $h=1$ の場合

64ビット加算器の場合



RCSAのビット数が漸次増加する。

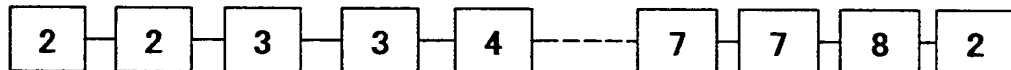
128ビット加算器の場合



RCSAのビット数が漸次増加する。

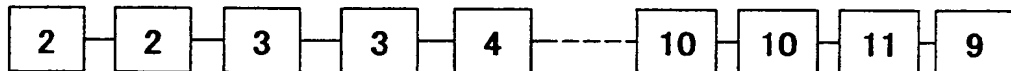
(f)初期値 $M_0=2$ 、公差 $h=0.5$ の場合

64ビット加算器の場合



RCSAのビット数が2個おきに漸次増加する。

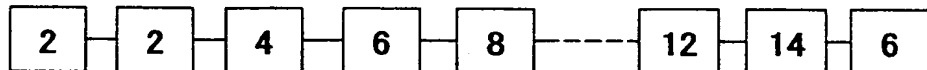
128ビット加算器の場合



RCSAのビット数が2個おきに漸次増加する。

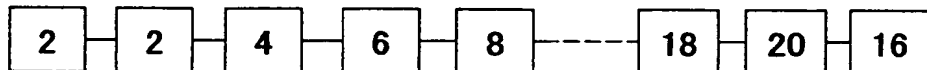
(g)初期値 $M_0=2$ 、公差 $h=2$ の場合

64ビット加算器の場合



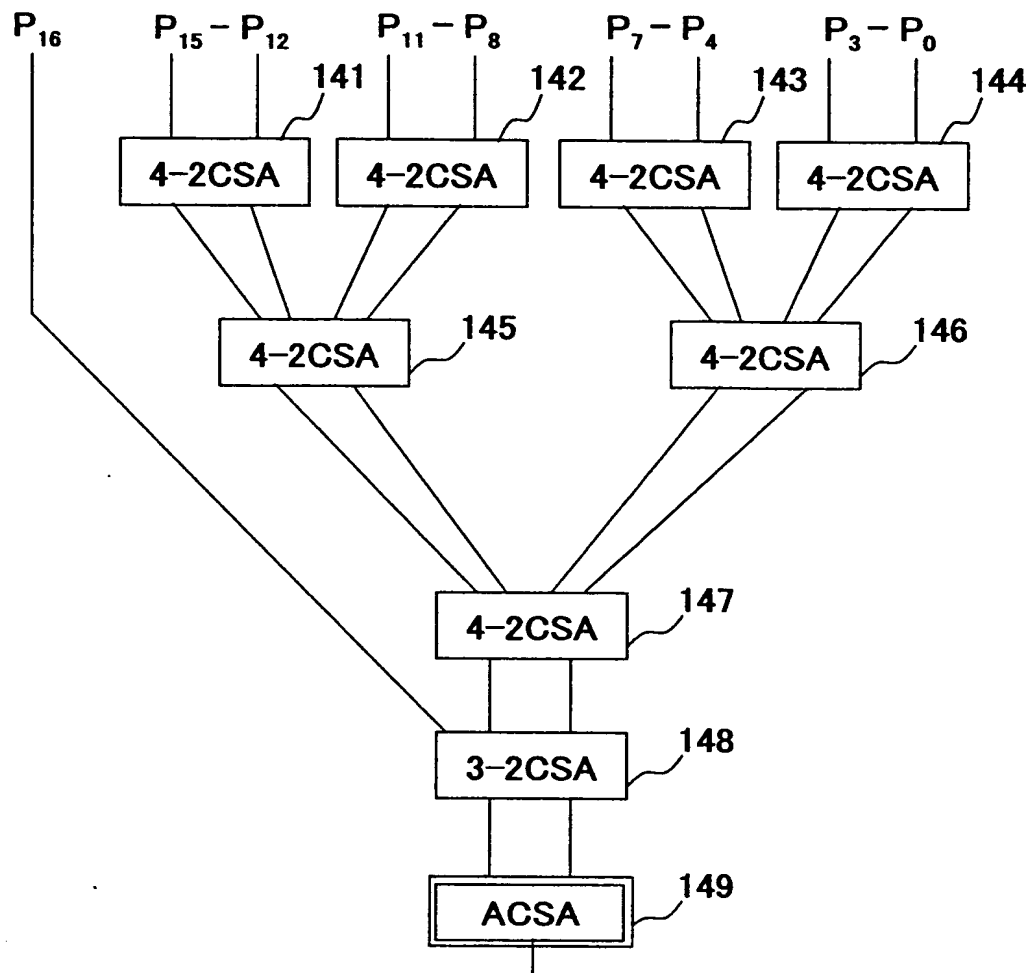
RCSAのビット数が2個ずつ漸次増加する。

128ビット加算器の場合



RCSAのビット数が2個ずつ漸次増加する。

【図 1 2】



【書類名】 要約書

【要約】

【課題】 暗号技術などにおける多ビットの加算に最適な、回路量を減らし、より高速としたキャリセレクト方式加算器加算器を提供する。

【解決手段】 2つのデータ入力と、2つのキャリ入力と、キャリ選択入力とを入力とし、2つのデータ入力のXOR出力を生成する第1 XOR素子と、このXOR出力を選択信号として第1キャリ入力と第1データ入力のいずれかを選択する第1マルチプレクサと、第2キャリ入力と第2データ入力のいずれかを選択する第2マルチプレクサと、キャリ選択入力を選択信号として2つのキャリ入力のいずれかを選択する第3マルチプレクサと、第3マルチプレクサの出力とXOR出力とのXOR出力を生成する第2 XOR素子とを有し、第1マルチプレクサの出力を第1キャリ出力とし、第2マルチプレクサの出力を第2キャリ出力とし、第3マルチプレクサの出力を加算値とする加算器。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 0 2 9 5]

1. 変更年月日 1 9 9 0 年 8 月 2 2 日

[変更理由] 新規登録

住 所 東京都港区虎ノ門1丁目7番12号

氏 名 沖電気工業株式会社